CLAIMS

What is claimed is:

1. A method for accessing data records from a large set of records stored in a database, comprising:

defining a plurality of boundaries to logically partition the large set of records into a plurality of buckets in accordance with a predefined sort order, said sort order based on data stored in at least two columns of a database table in which at least a base portion of each data record is stored;

in response to a request to retrieve a data record,

determining a bucket in which the data record is contained; and

retrieving a subset of the large set of records from the database corresponding to data records contained in the bucket.

2. The method of claim 1, wherein retrieving the subset of the large set of records comprises:

searching the plurality of boundaries for a sequentially-adjacent pair of boundaries as defined by the predetermined sort order that bound the data record in the request;

generating a query including a search clause defining a data set that is bounded by the sequentially-adjacent pair of boundaries; and

executing the query against the database.

3. The method of claim 1, further comprising storing a boundary marker for each boundary comprising data pertaining to said at least two columns of the database table.

4. The method of claim 3, wherein each boundary marker comprises first, second and tertiary data values pertaining to data values stored in respective columns in the database table, said data values pertaining to an actual record or pseudo record to which that boundary marker corresponds.

5. The method of claim 3, wherein the boundary markers are stored as a concatenated list.

6. The method of claim 3, wherein the boundary markers are stored as a set of respective records in the database.

7. The method of claim 1, wherein the subset of data records corresponding to the bucket is provided to a batch processing component, further comprising retrieving other buckets of data and providing them to the batch processing component until data records corresponding to the entire large set of data records have been provided to the batch processing component.

8. The method of claim 1, further comprising:

monitoring user navigation events;

sending data corresponding to a current viewset to a client, said viewset comprising a subset of data records in the data bucket and corresponding to a current user navigation context; and in response to a user navigation event corresponding to a request to navigate to a new navigation context, updating the current viewset with a new viewset by sending a new subset of data records in the data bucket, wherein the new viewset corresponds to the new navigation context.

9. The method of claim 8, further comprising:

determining if a user navigation event corresponds to a request to view one or more data records that are not in a current data bucket;

and, in response to such a determination,

formulating a query request corresponding to a new data bucket that contains the data records that are requested to be viewed;

executing the query to retrieve data records corresponding to the new data bucket from the database; and

providing data corresponding to a new viewset comprising a subset of the data records in the new data bucket to the client.

10. The method of claim 8, wherein the client is a web browser and the data corresponding to the viewsets are sent as HTML data over a network to a client machine on which the web browser is running.

11. A method for accessing data records from a database, comprising:

determining a sort order in which the data records are to be provided, said sort order including data from at least two columns of a database table from which the data records are based;

defining a plurality of boundaries, each boundary corresponding to a relative position of an actual or pseudo data record within a set of data records sorted in accordance with the sort order;

in response to a request for a data record or a group of related data records within the set of data records,

determining a lower and upper boundary corresponding to a data bucket containing the data record or at least a portion of the group of related data records, said lower and upper boundaries being selected from among the plurality of boundaries points; and

retrieving data records logically contained within the data bucket based on the relative positions of the lower and upper bucket boundaries within the sorted set of data records.

12 The method of claim 1, wherein each boundary is marked by a boundary marker comprising a unique set of data corresponding to said at least two columns in the database table;

13. The method of claim 12, further comprising storing the boundary markers in correspondence with the sort order.

14. The method of claim 11, wherein defining the boundaries comprises:

opening a data set corresponding to the data records, sorted in accordance with the sort order and containing data corresponding to at least said at least two columns in the database table; and

looping through the data set, defining boundaries every N rows in the data set.

15. The method of claim 14, wherein subsets of the data set are opened on a bucket-wise basis using an estimated set of boundaries, further comprising defining actual boundaries to replace the estimated boundaries by looping through each subset and defining actual boundaries every N rows of that subset.

16. The method of claim 11, wherein determining the lower and upper boundaries comprises:

searching through the plurality of boundaries to identify the boundary having a relative position in the sorted order that is closest to the data record or a first record in the group of related data records from among boundaries having relative positions prior to the data record or first record in the group of related data records in the sorted order to determine the lower boundary; and

identifying the boundary having the next relative position after the boundary determined for the lower boundary to determine the upper boundary.

17. The method of claim 11, further comprising:

monitoring a number of records returned by a query that is executed to retrieve the data bucket; and

altering the query to dynamically adjust the size of subsequent buckets based on the number of records returned by a prior query relative to a number of records contained in a data bucket having a desired size.

18. The method of claim 17, wherein the size of the subsequent buckets are adjusted by skipping a determined number of boundaries.

19. A machine-readable media on which a plurality of instructions are stored that when executed by a computing machine perform the operations of:

receiving a request to retrieve data records from a database on which a large set of records are stored;

searching a predefined set of boundaries that logically partition the large set of records into a plurality of buckets in accordance with a predefined sort order that is based on data stored in at least two columns of a database table in which at least a base portion of each data record is stored;

determining a bucket in which the data record is contained;

formulating a query to retrieve a subset of the large set of records, said subset corresponding to the data records contained in the bucket;

submitting the query to the database for execution; and

receiving the subset of records contained in the bucket from the database.

20. The machine-readable media of claim 19, wherein retrieving the subset of the large set of records contained in the bucket comprises:

searching the plurality of boundaries for a sequentially-adjacent pair of boundaries as defined by the predetermined sort order that bound the data record in the request; and

formulating the query to include a search clause defining a data set that is bounded by the sequentially-adjacent pair of boundaries.

21. The machine-readable media of claim 19, wherein execution of the instructions further performs the operations of:

providing the subset of data records corresponding to the bucket to a batch processing component running on the computing machine or on another computing machine linked in communication with the machine;

iteratively formulating and submitting queries to the database to retrieve , other buckets of data and providing them to the batch processing component until data records corresponding to the entire large set of data records have been provided to the batch processing component.

22. The machine-readable media of claim 19, wherein execution of the instructions further performs the operations of:

monitoring user navigation events received from a client;

sending data corresponding to a current viewset to the client, said viewset comprising a subset of data records in the data bucket and corresponding to a current user navigation context; and

in response to a user navigation event corresponding to a request to navigate to a new navigation context, updating the current viewset with a new viewset by sending a new subset of data records in the data bucket, wherein the new viewset corresponds to the new navigation context.

23. The machine-readable media of claim 22, wherein execution of the instructions further performs the operations of:

determining if a user navigation event corresponds to a request to view one or more data records that are not in a current data bucket;

and, in response to such a determination,

formulating a query request corresponding to a new data bucket that contains the data records that are requested to be viewed;

executing the query to retrieve data records corresponding to the new data bucket from the database; and

providing data corresponding to a new viewset comprising a subset of the data records in the new data bucket to the client.

24. The machine-readable media of claim 22, wherein the client is a web browser and the data corresponding to the viewsets are sent as HTML data over a network to a client machine on which the web browser is running.

25. The machine-readable media of claim 19, wherein execution of the instructions further performs the operations of:

monitoring a number of records returned by a query that is submitted to the database to retrieve the bucket of data records; and

altering the query to dynamically adjust the size of subsequent buckets based on the number of records returned by a prior query relative to a number of records contained in a bucket having a desired size.

26. The machine-readable media of claim 25, wherein the size of the subsequent buckets are adjusted by skipping a determined number of boundaries.

27. A machine-readable media on which a plurality of instructions are stored that when executed by a computing machine perform the operations of:

defining a plurality of boundaries to logically partition a large set of records into a plurality of buckets in accordance with a predefined sort order, said sort order based on data stored in at least two columns of a database table in which at least a base portion of each data record is stored;

storing a boundary marker for each boundary comprising a unique set of data pertaining to said at least two columns of the database table.

28. The machine-readable media of claim 27, wherein each boundary marker comprises first, second and tertiary data values pertaining to data values stored in respective columns in the database table, said data values pertaining to an actual record or pseudo record to which that boundary marker corresponds.

29. The machine-readable media of claim 27, wherein the boundary markers are stored as a concatenated list.

30. The machine-readable media of claim 27, wherein the boundary markers are stored as a set of respective records in the database..

31. The machine-readable media of claim 27, wherein the operation of defining the boundaries comprises:

opening a data set corresponding to the large set of data records, sorted in accordance with the sort order and containing data corresponding to at least said at least two columns in the database table; and

looping through the data set, defining boundaries every N rows in the data set.

32. The machine-readable media of claim 31, wherein subsets of the data set are opened on a bucket-wise basis using an estimated set of boundaries, further comprising defining actual boundaries to replace the estimated boundaries by

looping through each subset and defining actual boundaries every N rows of that subset.

33. A computer system comprising:

a memory in which a plurality of machine instructions are stored;

a network interface to link the computer in communication with a database server; and

a processor, coupled to the memory and the network interface, to execute the plurality of machine instructions to cause the computer system to perform the operations of:

receiving a request to retrieve data records from a database hosted by the database server on which a large set of records are stored;

searching a predefined set of boundaries that logically partition the large set of records into a plurality of buckets in accordance with a predefined sort order that is based on data stored in at least two columns of a database table in which at least a base portion of each data record is stored;

determining a bucket in which the data record is contained;

formulating a query to retrieve a subset of the large set of records, said subset corresponding to the data records contained in the bucket;

submitting the query over the network interface to the database server for execution; and

receiving the subset of records contained in the bucket from the database server.

34. The computer system of claim 33, wherein retrieving the subset of the large set of records contained in the bucket comprises:

searching the plurality of boundaries for a sequentially-adjacent pair of boundaries as defined by the predetermined sort order that bound the data record in the request; and

formulating the query to include a search clause defining a data set that is bounded by the sequentially-adjacent pair of boundaries.

35. The computer system of claim 33, wherein execution of the machine instructions further performs the operations of:

providing the subset of data records corresponding to the bucket to a batch processing component running on the computing machine or on another computing machine linked in communication with the machine;

iteratively formulating and submitting queries to the database to retrieve , other buckets of data and providing them to the batch processing component until data records corresponding to the entire large set of data records have been provided to the batch processing component.

36. The computer system of claim 33, wherein the computer system is linked in communication with a client machine via the network interface and wherein execution of the machine instructions further performs the operations of:

monitoring user navigation events received from the client machine;

sending data corresponding to a current viewset to the client machine, said viewset comprising a subset of data records in the data bucket and corresponding to a current user navigation context; and

in response to a user navigation event corresponding to a request to navigate to a new navigation context, updating the current viewset with a new viewset by sending a new subset of data records in the data bucket to the client machine, wherein the new viewset corresponds to the new navigation context.

37. The computer system of claim 36, wherein execution of the machine instructions further performs the operations of:

determining if a user navigation event corresponds to a request to view one or more data records that are not in a current data bucket;

and, in response to such a determination,

formulating a query request corresponding to a new data bucket that contains the data records that are requested to be viewed;

submitting the query request to the database server;

receiving corresponding to the new data bucket from the database server in response to the query request; and

providing data corresponding to a new viewset comprising a subset of the data records in the new data bucket to the client machine.

38. The computer system of claim 36, wherein a web browser is running on the client machine and the data corresponding to the viewsets are sent as HTML data via the network interface to the client machine.

39. The computer system of claim 33, wherein execution of the machine instructions further performs the operations of:

monitoring a number of records returned by a query that is submitted to the database server to retrieve the bucket of data records; and altering the query to dynamically adjust the size of subsequent buckets based on the number of records returned by a prior query relative to a number of records contained in a bucket having a desired size.

40. The computer system of claim 39, wherein the size of the subsequent buckets are adjusted by skipping a determined number of boundaries.